Using Association Rules to Make Rule-based Classifiers Robust

Hong Hu and Jiuyong Li
Department of Mathematics and Computing
The University of Southern Queensland
4350, Australia
huhong@usq.edu.au and jiuyong@usq.edu.au

Abstract

Rule-based classification systems have been widely used in real world applications because of the easy interpretability of rules. Many traditional rule-based classifiers prefer small rule sets to large rule sets, but small classifiers are sensitive to the missing values in unseen test data. In this paper, we present a larger classifier that is less sensitive to the missing values in unseen test data. We experimentally show that it is more accurate than some benchmark classifies when unseen test data have missing values.

Keywords: data mining, association rule, classification, robustness.

1 Introduction

Automatic classification has been a goal for machine learning and data mining, and rule based methods are widely accepted due to their easy understandability and interpretability.

Rule discovery has been studied for more than twenty years and a number of methods have been proposed. They are typically classified into the following categories.

1. Covering algorithm based methods: a covering rule discovery algorithm employs the divide and conquer approach and works in the following manner. A "best" rule is found first from a data set and then all records covered (or explained) by the rule is removed from the data set. This procedure repeated until there is no record left in the data set. The way to find the "best" rule is usually by some heuristics, e.g. entropy. Some typical methods in this category are AQ15 (Michalski, Mozetic, Hong & Lavrac 1986), CN2 (Clark & Niblett 1989, Clark & Boswell 1991) and CPA (Yin & Han 2003).

2. Decision tree based methods: a decision tree is also a typical divide and conquer approach. It differs from a covering rule discovery algorithm in that it divides the training data into disjoint sub data sets by some attribute values simultaneously. Those sub data sets are simultaneously divided by some other attribute values recursively until each sub data set contains records of one class, or nearly. The partitions are guided by some heuristic measures, e.g. information gain and information gain ratio (Quinlan 1993). Each path from the root to a leaf in the decision tree is interpreted as a rule. C4.5rules (Quinlan 1993) is a typical method in this category.

3. Association based methods: Association rules are proposed for resolving market basket problems on transactional data. However, when all rule targets are constrained by the class labels, association rules become class (or constraint) association rules and they can be used for classification purpose. All association rule mining algorithms, e.g. Apriori (Agrawal & Srikant 1994) and FP-growth (Han, Pei & Yin 2000), can be easily adapted to mining class association rules. Some typical association based classification methods are CBA (Liu, Hsu & Ma 1998) and CMAR (Li, Han & Pei 2001).

4. Association based optimal methods: the main characteristic of association rule mining is to use the upwards closure property of the support to confine the searching space. When the goal of the rule mining is to find high confidence (or accurate) rules as in classification applications. The problem becomes the optimal rule set discovery. The upwards closure property of confidence further confines the search space, and mining optimal class association rules is more efficient than mining association rules. Some typical optimal class association rule mining algorithms are PC optimality rule mining (Bayardo & Agrawal 1999) and optimal class association rule set mining (Li, Shen & Topor 2002).

In the above four types of methods, the first two types of methods usually produce small rule sets because the overlapping of the covered record sets of rules are minimised. In contrast, the latter two types of methods usually generate large rule sets because of the covered record sets of rules are highly overlapped. An association rule set is the largest. An optimal class association rule set is significantly smaller than an association rule set but is still large. It is not uncommon that an association rule set is 100 times larger than an optimal class association rule set while an optimal rule set is 100 times larger than a rule set from C4.5rules.

Small rule sets are preferred in building the traditional rule based classifiers. Raw rules interpreted from a decision tree significantly outnumbers rules in the final rule set from C4.5rules. These raw rules are pruned by Minimum Description Length Principle (MDLP) (Rissanen 1983). In the procedure of the pruning, most low support rules are removed. (This indirectly justifies the use of the minimum support in class association rule mining.) Even in an typical association rule based classifier, i.e. CBA, rules are pruned by a covering algorithm based post processing method. The classifiers from CBA is small too.

An argument for preferring small rule sets is that they do not overfit the training data sets and result higher accuracies in test data sets. (We will dispute this brief in the discussion.) However, small classifiers embed some problems. Before we discuss these problems, we will have a look at classifiers.

Rule based classification usually involves two stages, learning and test. Consider a relational data set where each record is assigned a category (class), called a training data set. In the learning stage, we generate a rule set where each rule associates a pattern with a class. Then in the test stage, we apply this rule set to test data without class information, and to predict the class that a record in the test data set belongs to. If the predictive class is the class that the record supposed to belong to, then the prediction is correct. Otherwise, it is wrong. The proportion of correct predictions from test data is accuracy.

Classifiers refer to a rule set and the mechanism for its making predictions. Highly accurate classifiers are generally preferred.

There are roughly two types of models for building rule based classifiers.

1. Ordered rule based classifiers: rules are organised as a sequence, e.g. in the descending accuracy order. When classifying a coming record, the first matching rule in the sequence makes the prediction. This sequence is usually tailed by a default class. When there is no rules in the sequence matches the coming record, the class of the record is predicted as the default one. C4.5rules (Quinlan 1993) and CBA (Liu et al. 1998) employ this model.

2. Unordered rule based classifiers: rules are not organised in a sequence and all (or most) matching rules participate the determination of the class of a coming record. A straightforward way is to accept the majority vote of rules like in CPAR (Yin & Han 2003). A more complex method is to compare the actual accuracies obtained from the multiple rules for all possible classes. The one getting the highest accuracy will be the final prediction. Improved CN2 (Clark & Boswell 1991) and CMAR (Li et al. 2001) employ this method.

We do not discuss committee prediction, e.g. Bagging (Breiman 1996) and Boosting (Freund & Schapire 1996, Freund & Schapire 1997), which use multiple classifiers.

The first model is simple and effective whereas the second model is not yet to be mature. The first model makes a prediction based on the most likelihood. This is because that rules with higher accuracies usually precede rules with lower accuracies and the accuracy approximates the conditional probability when data set is large. However, an important concept for the second model, independence of rules, is yet fully developed. Clark gave a practical definition (theoretical justification was not provided) for the independence and used the model in the improved CN2. Interestingly, a rule set from a covering algorithm based method does not actually support the second model. Each record is supposed to support only one rule in the rule set and therefore it is rare that a prediction makes use of multiple rules. CMAR makes use of all class association rules that do support the second model. However, it does not address the problem of the independence of rules in an association rule set where the majority of rules are correlated and their covered record set is highly overlapped.

in contrast, ordered rule based classifiers are relatively effective and stable. However, they have some other problems.

Small classifiers do not tolerate missing values in the unseen test data and hence are not robust. All ordered rule based classifiers employ small rule sets. Rule sets from C4.5rules are very small since MDLP is used for post pruning. CBA makes use of large association rule sets but prunes them to small rule sets by a covering algorithm. The small rule sets are too slim to tolerate the possible missing values in the unseen test data. For example, two rules, "Test A = high ⇒ diabetes" and "Test B = high and Symptom = C ⇒ diabetes", accounts for a group of patients. Currently used small classifiers include only one rule, say the first rule. Therefore, patients who do not take Test A but take Test B and have symptom C will miss the matching of the first rule, and may be classified as normal by the default prediction.

Small classifiers rely significantly on the default prediction, and the predictions based on the default prediction may be misleading. For example, in data set Hypothyroid, 95.2% records belong to class Negative and only 4.8 % records belong to class Hypothyroid. So, if we set the default prediction as Negative, then a classifier that has no rule will give 95.2% accuracy. You can see that how accuracy is floated by the default prediction. Further, this distribution knowledge is too general to be useful. For example, a doctor uses his patient data to build a rule based diagnosis system. 95% patients coming to him are healthy, and hence the system sets the default as healthy. Though the default easily picks up 95% accuracy, this accuracy is meaningless for the doctor.

In this paper, we will make use of optimal class association rules to build more robust classifiers. A side product of this work is to limit the use of default prediction in the prediction.

2 Robust rule based predictions

In practice, a rule set is generated from the history data and is used to make predictions on future coming data. Usually, the volume of the history data is huge and hence partial typical history data are used to generate the rule set. It is common that the future coming data are not as complete as those typical training data, e.g. some attribute values in a record are missing. The goal of robust prediction is to find a rule set to make reasonable highly accurate predictions even when the test data set is not as complete as the training data.

For example, in a medical application, the training data is the set of classified history data. They are usually selected typical complete records. A test record is the one for a coming patient. In many cases, this record is not yet complete. It is desirable that a rule set can make a certain reasonable prediction for such incomplete record.

There is a common case for missing values in test data. When some cases are under study, all sort of information is collected. However, in practice, only partial information is available. As a result, records for general cases are less complete than that for the typical cases.

Robustness has twofold meanings in terms of dealing with missing values. The toleration of missing data in training data is one, and the toleration of missing data in test data is the other. There are some research for handing missing data in training data (Clark & Niblett 1989, Mingers 1989, Quinlan 1993, Batista & Monard 2003), but no research on handling missing data in test data for rule based prediction apart from our previous work (Li, Topor & Shen 2002). In this paper, we focus on the later and evaluate a robust optimal class association rule based classifier.

This work is different from other existing methods for handling missing data. General methods for handling missing values are to pre-process data by substituting missing values by estimations using some approaches, e.g. the nearest neighbours (Batista & Monard 2003). In this paper, we do not estimate and substitute any missing values, but to make use of larger rule sets to make classifiers be "immune" from the missing test data.

3 Ordered rule based classifier and optimal rule sets

An intuitive solution for robustness is the redundancy. We have a look at an example in the telecommunication. To ensure the data transfer has minimum errors caused by missing or mistake bits. Some redundant bits are used to make up.

In this paper, we also use the redundant rules to make a rule based classifier more robust. When some rules are paralysed by missing values, alternative rules may make up partial predictions. For example, if a classifier keeps both rules "Test A = high ⇒ diabetes" and "Test B = high and Symptom = C ⇒ diabetes", then it will not misclassify patients who do not take test A.

The question is that which redundant rules we should use to make a rule based classifier robust from a huge number of rules. We had a theoretical framework for constructing robust predictive rule sets in (Li, Topor & Shen 2002), and in this paper we design and evaluate a practical robust rule based classifier. The rule base for this classifier is a 1-optimal rule set discussed in the following.

3.1 Ordered rule based classifiers

We have a look at how ordered rule based classifiers work. In the rest of this paper, "classifier" means an ordered rule based classifier since we do not study unordered rule based classifier in this paper.

We first present some useful definitions and notations we use in the paper. Let D be a relational data set with n attributes, and T be a record containing a set of attribute-value pairs. A *pattern* P is a subset of T. The *support* of a pattern P is the ratio of the number of records containing P to the number of records in the data set, denoted by $sup(P)$. A *rule* is in form of $P \Rightarrow c$, where c is a class. The support of rule $P \Rightarrow c$ is $sup(Pc)$. where Pc is a short for $P \cup c$. The confidence of the rule is $sup(Pc)/sup(P)$, denoted by $conf(P \Rightarrow c)$. Rules we discussed in this paper are strong rules e.g. their support and confidence are above the minimum support and confidence respectively.

In the practice of rule based classification, a set of rules is usually sorted by decreasing accuracy, and tailed by a default prediction. This ordered rule set is called rule based classifier. In classifying an unseen test record (an input record has no class information), the first rule that matches the record classifies it. If no rule matches the record, the default prediction is used.

We do not know the accuracy of rules before they are tested. However, we need to know their accuracy before we test them since we have to order them in the classifier. Therefore, we need to estimate the accuracy of rules first.

There are a few methods in estimating rule accuracy. Laplace accuracy is a widely used estimation. We rewrite the Laplace accuracy in terms of support and cover set as follows.

$$acc(A \Rightarrow c) = \frac{sup(A \Rightarrow c) \times |D| + 1}{|cov(A \Rightarrow c)| + |C|}$$

where $|C|$ is the number of all classes, $sup(A \Rightarrow c) \times |D|$ is the number of correct predictions made by the rule on training data and $|cov(A \Rightarrow c)|$ is the number of total predictions made by the rule when no other rules are used.

An estimated accuracy of hypothesis is presented in (Mitchell 1997). Quinlan used pessimistic error rate in rule pruning (Quinlan 1993).

It is not our intention to argue which estimation is best in this paper. Whatever estimation will not change the main conclusions of this paper. We use Laplace accuracy in the experiments.

3.2 From optimal rule sets to 1-optimal rule sets

We note that all rules are sorted by their accuracies in a classifier. Some rules are never used in the predictions. For example, given two rules $a \Rightarrow z$ and $ab \Rightarrow z$ (we simplify the attribute and value pair as a letter here.), Assume that the first rule is more accurate than the second rule. The first rule will precede the second rule in a classifier, and the second rule will never be used since all records matching the second rule will match the first rule too. In practice, you never see the second rule in a classifier.

In general, only those more general and accurate rules are possibly in a classifier. Here, we say a rule is more general if it contains partial conditions of a more specific rule. We call a set of those more general and accurate rules as the optimal rule set (Li, Topor & Shen 2002). All rules for building a classifier have to be from the optimal rule set. The optimal rule set is the source for choosing redundant rules.

It is possible to build an optimal classifier by using all rules in the optimal rule set, and this optimal classifier is presumed to be the most robust rule set since it includes all rules we can use. However, this rule set is usually very large. We consider the following way to simplify the optimal classifier.

The simplest optimal rule set will be a set of the most accurate rules covering every record in the training data. We call it the min-optimal rule set. The min-optimal rule set is usually bigger than a traditional classification rule set, e.g. a rule set from C4.5rules (Quinlan 1993). C4.5rules uses Minimum Description Length Principle (MDLP) (Rissanen 1983) method to further simplify a rule set from a decision tree. Therefore, there are some redundant rules in the min-optimal classifier, and the min-optimal classifier should be more robust than a traditional classifier.

We may go further to include more redundant rules in a classifier. In the min-optimal rule set, we include only the most accurate rule for a record. We may include the first two most accurate rules for a record. The problem is that these two rules may be highly overlapping, for example, rule $ab \Rightarrow z$ and rule $bc \Rightarrow z$. The missing value b will paralyse both rules, and this is not good for the overall robustness. Therefore, we require these two rules to be disjunctive. However, a record may not support two disjunctive rules. In this case, we will have to include more rules to create disjunction. For example, given rules $ab \Rightarrow z$, $bc \Rightarrow z$, and $cd \Rightarrow z$. no matter a, b, c, or d is missing, one rule still works. Based on this idea, we create 1-optimal rule set that includes at least two rules for every record in the training data. The 1-optimal rule set includes more redundant rules than the min-optimal rule set, and hence is more robust.

The precise definitions for min-optimal, 1-optimal and optimal rule sets and their robustness relationships are given in (Li, Topor & Shen 2002). The main conclusions are listed as follows. The optimal rule set is more robust than the 1-optimal rule set and both are more robust than the min-optimal rule set. 1-optimal rule set get a good tradeoff for rule size and robustness.

4 Building optimal association classifiers (OAC)

Optimal association classifiers are based on 1-optimal rule sets. The key for obtaining 1-optimal association rule is to obtain the optimal class association rule set. There are three ways to obtain it.

1. Association rule mining approach, e.g. Apriori (Agrawal & Srikant 1994) or FP-growth (Han et al. 2000), plus post-pruning,

2. Constraint association rule mining approach (Bayardo, Agrawal & Gunopulos 2000), and

3. optimal class association rule mining approach (Li, Shen & Topor 2002).

The first one is the most inefficient way among the three approaches since an association rule set is usually very large. In some cases, it is impossible to generate all association rules when the minimum support is set low in dense data sets.

The second one can be adapted to mine the optimal class association rule set when the minimum confidence improvement is set as zero. A major shortcoming for the second method is that it assumes the target is fixed for one class. Therefore, we need to discover optimal rule sets for every class first and then union them as the optimal rule set. When the number of classes is large, this involves certain redundant computation.

The third one is proposed for mining the optimal class association rule set. It is significantly faster than Apriori as shown in (Li, Shen & Topor 2002). It also uses less memory than Apriori since it does not generate all class association rules. It generates the optimal class association rule set with respect to all classes once and does not involve redundant computation as the second method. We employed the third method in our experiment.

Assume that we have the optimal rule class association rule set already. We consider how to build optimal association classifiers in the following.

Given a rule r, let $Attr(r)$ be the set of attributes whose values appear in the antecedent of r.

Algorithm 1 *Build optimal association classifiers (OAC)*

```
    Input: Data set $D$ and optimal rule set $R_o$
    Output: Optimal association classifier $C_o$

   // Select 1-optimal class association rule set
1      set $R = \emptyset$
2          for each record $T_i$ in $D$
3              set $R_i = \emptyset$ and let $R'_i$ include all rules covering $T_i$
4          select the most accurate rule $r$ in $R'_i$ and move it to $R_i$
5          let $A = Attr(r)$
6          while ($A \neq \emptyset$ AND $R'_i \neq \emptyset$)
7              select the most accurate rule $r'$ in $R'_i$ and move it to $R_i$
8              let $A = A \cap Attr(r')$
9          let $R = R \bigcup R_i$

   // Build optimal association classifiers
10     initiate $C_o$ be an empty sequence
11     while($D \neq \emptyset$ AND $R \neq \emptyset$)
12         select rule $r$ in $R$ making smallest errors on $D$
13         remove $r$ from $R$ and append it to $C_o$
14         remove all records covered by $r$ from $D$
15     sort rules in $R$ by accuracy
16     append $R$ to $C_o$
17     return $C_o$
```

There are two stages in the above algorithm.

The first stage is to select a 1-optimal class association rule set. Rules are selected record by record. In line 3, all rules covering a record is put in rule set R'_i. Line 4 selects the most accurate rule and moves it from R'_i to R_i. Line 6 and 7 select other most accurate rules in R'_i to complement rules in R_i until conditions of all rules are disjointed or there is no rules left in R'_i. You can see that for T with any one missing value rules in R_i are still able to make prediction on it. This is the name of 1-optimal comes from. All 1-optimal rules for every record are put together to form 1-optimal rule set.

The second stage is to build an optimal association classifier (OAC). We construct OAC by selecting rules in a 1-optimal rule set by the covering algorithm. We recursively move a rule with the smallest misclassification rate to the classifier and remove its covered records in the training data set. When there is no record left, the remaining rules in the 1-optimal rule set are appended to the classifier in the order of accuracy. The default prediction is set as the majority class in training data set.

The building OAC is very similar to that for CBA (Liu et al. 1998) except the following two aspects. The input rule set for OAC is a 1-optimal class association rule set whereas the input rule set for CBA is an association rule set. After rules with the smallest misclassification rate are selected to the classifier, all remaining rules are appended to the OAC whereas they are discarded by CBA. Actually, these rules make the OAC more robust.

The time complexity of the above algorithm is $O(l|R'_i||D| + |R||D|)$ (we did not simplify it to keep it clear), where l is the number of conditions of the longest rules. Usually, $l \leq 10$, $|R'_i| < 50$ and $|R| < 500$. Therefore, this procedure is very efficient.

5 Experimental results

The main idea for the robustness is to use a rule set on a data set that is less complete than the data set which the rule set is from. In other words, test data has more missing values than training data.

Following the common practice, we employ 10 fold cross validation to separate training data and test data.

We generate a rule set from a normal training data set and test it on a test data set with added missing values. The missing values that we add to the test data are on top of possible missing values in the test data.

We use the following way to add more missing values to the test data. We randomly omitted some values in the test data sets to produce l-incomplete test data sets. When generating l-incomplete data sets, we control the total number of missing values, such that every record in the test data has l missing values on average.

To ensure that the accuracies from l-incomplete test data sets are reliable, we test every rule set on ten randomly generated incomplete test data sets and report the average. Consider a test data set is a portion in the 10 fold cross validation. A test accuracy of a data set is the average of 100 tests.

To ensure the reliability of the results, we choose 28 widely used data sets from UCI ML Repository (Blake & Merz 1998). They are: Anneal, Australian, Auto, Breast, Cleve, Crx, Diabetes, German, Glass, Heart, Hepatitis, Horse-colic, House-vote, Hypo, Ionosphere, Iris, Labor, Led7, Lymph, Mushrooms, Pima, Sick, Sonar, Tic-tac, Vehicle, Waveform and WineZoo. A brief description of them is in Table 1

Data set	Size	#Attr	#Class	Rule Size		
				C4.5rules	CBA	OAC
Anneal	898	38	5	21	35	78
Australian	690	14	2	8	159	510
Auto	205	25	7	27	61	105
Breast-cancer	699	10	2	10	47	83
Cleve	303	13	2	14	76	225
Crx	690	15	2	11	154	471
Diabetes	768	8	2	10	44	112
German	1000	20	2	35	293	994
Glass	214	9	7	11	31	41
Heart	270	13	2	10	43	157
Hepatitis	155	19	2	8	39	104
Horse-colic	368	22	2	6	115	302
House-vote	435	16	2	6	48	99
Hypo	3163	25	2	7	35	107
Ionosphere	351	34	2	12	52	123
Iris	150	4	3	3	5	9
Labor	57	16	2	5	17	26
Led7	3200	7	10	32	45	247
Lymph	148	18	4	9	43	72
Mushrooms	8124	22	2	16	37	94
Pima	768	8	2	10	44	112
Sick	2800	29	2	10	58	170
Sonar	208	60	2	10	51	160
Tic-tac	958	9	2	18	32	574
Vehicle	846	18	4	46	149	597
Waveform	5000	21	3	204	651	2843
Wine	178	13	3	8	11	23
Zoo	101	16	7	9	9	11
Average				21	85	302

Table 1: Data set and classifier size

The comparison targets are C4.5rules and CBA. We choose them based on the following reasons.

Firstly, C4.5rules is a benchmark rule based classifier in machine learning community and CBA is a benchmark rule based classifier in data mining community.

Secondly, they both use the ordered rule based classifier. OAC is based on ordered rules too. The ordered rule based classifier is simple and effective and therefore is widely used.

Our evaluation objectives are listed as follows: to demonstrate that OAC is more robust than C4.5rules and CBA; to demonstrate that OAC relies less on the default prediction than C4.5rules does.

To achieve our goals, we compare the accuracy of different classifiers on test data with increasing missing value level.

In the experiments, we use *local support* of rule $A \Rightarrow c$, which is $sup(Ac)/sup(A)$, to avoid too many rules in the large distributed classes and too few rules in the small distributed class. For example, in data set Hypothyroid, 95.2% records belong to class Negative and only 4.8 % records belong to class Hypothyroid. So, 5% (global) support is very small for class Negative, but is too large for class Hypothyroid.

The parameters for the optimal rule set generation are listed as follows. Local minimum support, 0.01, minimum confidence, 0.5, and maximum length of rules, 6. For both C4.5rules and CBA, we used their default settings.

We first have a look at the overall results.

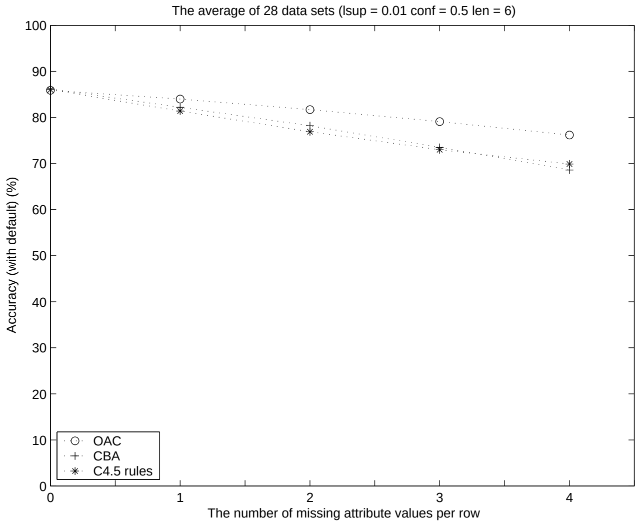

Figure 1: The average accuracy of three classifiers with the default on 28 data sets

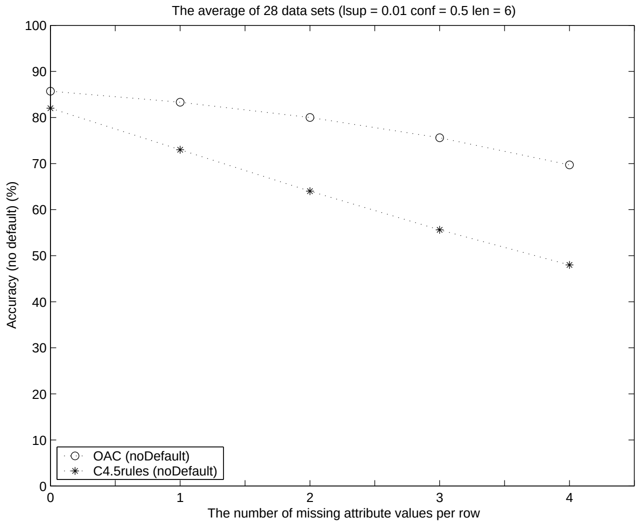

Figure 2: The average accuracy of two classifiers without default on 28 data sets

An optimal association classifier is 15 times larger than a C4.5rules classifier and 4 times larger than a CBA classifier on average, see Table 1. OAC makes use of larger rule sets than C4.5rules and CBA do. However, The rule sets used by OAC are at least 100 times smaller than association rule sets. Therefore, how to select these smaller association rule sets for prediction is not a trivial task.

The average accuracies of OAC, CBA and C4.5rules for 28 data sets on increasing missing value level are listed in Figure 1. The accuracy of OAC is higher than both C4.5rules and CBA when the test data is incomplete. Therefore, OAC is more robust than CBA and C4.5rules.

To show how OAC does not rely significantly on the default prediction. We drop the default prediction in classifiers, and a record matching no rules will be counted as an error. When there is no default prediction, the test accuracy ranges from 0 to 100%. In contrast, the test accuracy ranges from b to 100% when there is the default. The b is determined by the distribution of the default prediction. In some skewed data sets, b can be very big.

The average accuracies of OAC and C4.5rules without the default prediction for 28 data sets on increasing missing value level are listed in Figure 2. We did not compare with CBA since we could not drop the default prediction for CBA. When there is no default prediction in the classifiers, OAC is significantly more accurate than C4.5rules. By comparing Figure 2 and Figure 1, it is clear that OAC relies significantly less on the default prediction than C4.5rules.

More detailed results are listed in Table 2, 3, 4, 5 and 6.

Let us have a look at the results on complete test data set, Table 2. OAC nearly gains no accuracy improvement by using the default prediction. The function for the default prediction is largely replaced by those redundant rules. On average, OAC is slightly less accurate than C4.5rules and CBA with the default prediction. This means that those redundant rules do not do a good job as the simple default prediction does when the test data set is as complete as the training data. We did obtain similar accuracy as CBA when we kept classifiers as large as CBA (min-optimal association classifiers in our terms).

However, redundant rules do excellent jobs when the test data is not as complete as the training data as shown in the following tables. Remember shortcomings for the default prediction mentioned in Introduction. OAC without default prediction is a good choice for its accuracy and robustness.

datasets	C4.5rules		CBA	OAC	
	NoDefault	Default	Default	NoDefault	Default
Anneal	90.3	93.5	96.2	96.9	96.9
Australian	84.1	86.7	84.1	84.1	84.1
Auto	75.1	78.0	80.1	78.5	78.5
Breast-cancer	92.7	95.1	95.1	95.1	95.1
Cleve	77.3	80.5	80.2	81.9	81.9
Crx	83.8	86.4	85.5	83.5	83.5
Diabetes	71.1	76.7	75.9	75.9	75.9
German	66.6	73.8	73.0	73.0	73.0
Glass	63.6	72.5	76.3	73.5	73.9
Heart	80.0	83.0	81.1	81.1	81.1
Hepatitis	76.0	82.5	83.9	83.2	83.9
Horse-colic	79.6	83.4	81.3	83.1	83.1
House-votes	95.2	95.7	94.0	92.2	92.2
Hypo	99.2	99.3	98.8	98.6	98.6
Ionosphere	88.6	92.9	93.4	92.3	92.9
Iris	93.3	93.3	93.3	94.0	94.0
Labor	64.7	89.0	87.3	85.3	87.3
Led7	73.2	73.2	69.4	74.0	74.0
Lymph	73.1	78.4	86.5	81.8	81.8
Mushroom	99.9	99.9	100.0	100.0	100.0
Pima	71.3	78.0	76.2	78.1	78.1
Sick	97.5	97.9	97.2	97.3	97.3
Sonar	74.5	80.8	78.4	78.4	78.4
Tic-tac	97.3	99.5	98.1	98.2	98.2
vehicle	67.3	71.9	72.7	72.0	72.0
Waveform	72.8	75.7	82.0	81.7	81.7
Wine	94.9	97.7	97.1	96.5	97.1
Zoo	92.1	92.1	96.1	89.3	89.3
Average	82.0	86.0	86.2	85.7	85.8

Table 2: Accuracy for complete test data

6 Discussion

In this section, we will argue for a large rule set in classification application.

The results of this work seem directly against the bias for traditional classification rule learning. A simple rule set fitting the training data set is preferred in traditional rule learning because a small fitting rule set usually provides higher accurate predictions on unseen test data than a large fitting rule set. A large rule set overfits the training data.

Let us have a look at why overfitting problem happens. Consider the following two rules from a data set when there is no the minimum support requirement.

1. If outlook is sunny, temperature is mild and the wind is strong, then play tennis (1/14, 100%), and

2. If outlook is sunny, temperature is mild and the wind is weak, then do not play tennis (1/14, 100%).

These rules link strong wind with playing and weak wind with not playing. This is not a case by common sense. These two rules overfit the data since they only explain themselves.

datasets	C4.5rules		CBA	OAC	
	NoDefault	Default	Default	NoDefault	Default
Anneal	82.8	92.4	89.7	95.8	95.8
Australian	77.7	83.9	81.8	82.6	82.7
Auto	67.2	73.0	76.9	77.5	77.8
Breast-cancer	88.9	94.5	95.0	95.0	95.0
Cleve	69.5	77.9	77.9	81.9	81.9
Crx	77.1	84.0	82.2	82.7	82.8
Diabetes	57.7	73.9	71.9	73.8	73.8
German	60.3	71.7	71.0	72.4	72.8
Glass	48.1	62.7	65.8	66.9	70.0
Heart	69.9	79.0	77.9	80.7	80.7
Hepatitis	73.6	82.7	83.1	81.0	81.7
Horse-colic	76.0	81.7	79.8	82.1	82.1
House-votes	89.1	92.1	92.0	91.8	92.1
Hypo	97.9	98.9	98.4	98.4	98.4
Ionoshere	86.1	92.6	93.5	92.2	92.9
Iris	66.9	72.9	83.3	83.4	86.3
Labor	57.8	88.7	83.6	84.3	86.3
Led7	46.1	50.1	44.3	60.0	61.4
Lymph	68.5	77.0	84.7	81.5	81.5
Mushroom	93.3	96.7	99.0	99.5	99.9
Pima	57.1	74.3	71.9	76.6	76.6
Sick	95.5	97.6	96.3	97.0	97.0
Sonar	69.8	79.3	79.0	77.6	77.6
Tic-tac	69.4	83.3	89.7	87.9	89.9
vehicle	58.7	66.4	70.7	71.3	71.4
Waveform	69.6	73.9	81.2	81.0	81.0
Wine	85.9	91.0	91.3	95.8	96.4
Zoo	85.2	87.2	91.6	83.0	86.4
Average	73.0	81.4	82.3	83.3	84.0

Table 3: Accuracy for test data with one missing value

The overfitting is caused by rules fitting the noise data. It is not a direct result of large rule sets. Setting the minimum support is a way to avoid rules fitting noisy data.

When rules have low support, the fitting rule set is large. When rules have high support, the fitting rule set is small. This is where the belief that a large rule set overfits data comes from. If the minimum support is suitable, a large rule set will not overfit a data set. Surely, the determination of a best minimum support is not easy. However, keeping a large rule set is not the cause for decreasing accuracy on the test data.

We do observe that larger optimal rule sets predict less accurately than smaller min-optimal rule sets do, such as in Wine and Zoo data sets. We also observe the opposite phenomenon, such as in Labor and Lymph data sets. All these data sets are small data sets, and therefore we would rather say these are caused by variations.

Another argument in favor of for simplicity is that a simple rule set is more understandable. However, understanding a rule set is not as important as understanding predictions the rule set makes. If a rule set provides predictions with the most suitable rules, then its predictions are understandable. As to the rule set size, it does not matter whether it is large or small because rules are manipulated by a computer. In this sense, a large rule set is preferred since it includes more rules for variant situations.

7 Conclusions

In this paper, we discussed how to build robust rule based classifiers to predict on the test data that is not as complete as the training data. We make use of 1-optimal class association rule sets and build optimal association classifiers (OAC) that are larger than some conventional rule based classifiers. We use extensive experiments to demonstrate OAC is more robust than two benchmark rule based classifiers, C4.5rules and CBA. The experimental results

datasets	C4.5rules		CBA	OAC	
	NoDefault	Default	Default	NoDefault	Default
Anneal	74.2	90.5	83.1	94.5	94.7
Australian	70.3	80.6	79.6	80.6	81.2
Auto	61.8	69.3	73.1	76.0	76.3
Breast-cancer	81.5	92.4	94.3	94.7	94.8
Cleve	59.7	73.9	74.3	81.0	81.3
Crx	70.2	81.9	79.3	81.0	81.4
Diabetes	43.5	70.8	66.5	72.7	73.3
German	54.0	70.0	68.8	70.5	72.4
Glass	35.4	53.2	53.1	58.3	63.9
Heart	57.6	73.7	72.4	77.8	78.0
Hepatitis	70.4	81.7	82.1	78.7	79.6
Horse-colic	72.2	80.3	77.6	80.6	81.4
House-votes	82.6	89.4	89.1	91.3	92.1
Hypo	96.3	98.5	98.2	98.2	98.2
Ionoshere	83.2	92.1	93.6	92.2	93.0
Iris	41.7	54.5	69.6	68.0	74.7
Labor	48.9	85.9	86.0	78.5	83.7
Led7	24.9	31.8	27.1	45.1	48.3
Lymph	62.4	74.5	83.7	80.4	80.4
Mushroom	86.6	93.4	97.8	98.6	99.6
Pima	43.3	71.2	66.9	73.5	73.9
Sick	92.9	97.3	95.1	96.6	96.6
Sonar	64.8	77.0	78.4	76.7	77.4
Tic-tac	46.8	70.4	83.1	74.0	82.4
Vehicle	49.9	60.3	67.9	70.8	71.3
Waveform	65.3	71.6	80.1	80.6	80.7
Wine	76.2	83.8	86.2	93.3	94.2
Zoo	75.9	82.1	83.3	76.3	82.1
Average	64.0	76.9	78.2	80.0	81.7

Table 4: Accuracy for test data with two missing values

datasets	C4.5rules		CBA	OAC	
	NoDefault	Default	Default	NoDefault	Default
Anneal	67.0	89.7	75.2	92.1	93.1
Australian	63.4	78.0	73.4	77.5	79.8
Auto	55.1	64.2	68.1	73.2	73.9
Breast-cancer	72.6	90.4	92.7	93.9	94.2
Cleve	50.3	69.8	71.0	77.9	79.3
Crx	62.6	79.2	75.9	78.5	80.4
Diabetes	29.7	68.3	58.2	69.8	72.0
German	47.6	67.9	65.0	67.9	72.5
Glass	26.0	49.8	39.6	47.8	59.5
Heart	44.6	70.0	66.9	76.2	76.7
Hepatitis	65.5	81.4	78.6	79.0	81.4
Horse-colic	67.8	78.3	74.8	78.0	80.2
House-votes	75.6	86.8	86.9	89.9	91.5
Hypo	94.4	98.1	97.5	98.1	98.2
Ionoshere	79.9	91.6	93.2	91.8	92.7
Iris	15.6	33.6	51.8	38.5	52.8
Labor	42.8	84.3	83.2	73.6	81.6
Led7	11.1	19.7	16.8	29.0	34.1
Lymph	57.1	72.8	78.7	78.7	79.0
Mushroom	80.0	90.4	96.4	97.5	99.2
Pima	29.3	68.2	58.9	68.8	71.5
Sick	90.5	97.1	93.4	96.4	96.6
Sonar	58.4	75.0	77.9	76.6	78.4
Tic-tac	29.6	60.4	76.6	57.7	77.3
vehicle	43.0	55.7	64.7	68.6	69.7
Waveform	61.0	68.8	79.0	79.5	79.7
Wine	67.2	78.3	82.9	89.1	91
Zoo	69.4	77.7	80.7	71.5	79.5
Average	55.6	73.0	73.5	75.6	79.1

Table 5: Accuracy for test data with three missing values

also show that OAC does not significantly rely on the default prediction, and this makes predictions more understandable. Given the frequent missing values in real world data sets, OAC has great potential in building robust classifiers in the future applications.

Acknowledgement

We sincerely think Ross Quinlan for making C4.5rules available for research, and Bing Liu for providing CBA for the comparison.

References

Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules in large databases, *in* 'Proceedings of the Twentieth International Conference on Very Large Databases', Santiago, Chile, pp. 487–499.

Batista, G. E. A. P. A. & Monard, M. C. (2003), 'An analysis of four missing data treatment methods for supervised learning', *Applied Artificial Intelligence* **17(5-6)**, 519–533.

Bayardo, R. & Agrawal, R. (1999), Mining the most interesting rules, *in* 'Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, N.Y., pp. 145–154.

Bayardo, R., Agrawal, R. & Gunopulos, D. (2000), 'Constraint-based rule mining in large, dense database', *Data Mining and Knowledge Discovery Journal* **4(2/3)**, 217–240.

Blake, E. K. C. & Merz, C. J. (1998), 'UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html'.

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**, 123–140.

Clark, P. & Boswell, R. (1991), Rule induction with CN2: Some recent improvements, *in* 'Machine Learning - EWSL-91', pp. 151–163.

Clark, P. & Niblett, T. (1989), 'The CN2 induction algorithm', *Machine Learning* **3**(4), 261–283.

Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, *in* 'International Conference on Machine Learning', pp. 148–156. *citeseer.nj.nec.com/freund96experiments.html

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences* **55(1)**, 119–139.

Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, *in* 'Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00)', May, pp. 1–12.

Li, J., Shen, H. & Topor, R. (2002), 'Mining the optimal class association rule set', *Knowledge-Based System* **15(7)**, 399–405.

Li, J., Topor, R. & Shen, H. (2002), Construct robust rule sets for classification, *in* 'Proceedings of the eighth ACMKDD international conference on knowledge discovery and data mining', ACM press, Edmonton, Canada, pp. 564 – 569.

Li, W., Han, J. & Pei, J. (2001), CMAR: Accurate and efficient classification based on multiple class-association rules, *in* 'Proceedings 2001 IEEE International Conference on Data Mining (ICDM 2001)', IEEE Computer Society Press, pp. 369–376.

Liu, B., Hsu, W. & Ma, Y. (1998), Integrating classification and association rule mining, *in* 'Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)', pp. 27–31.

Michalski, R., Mozetic, I., Hong, J. & Lavrac, N. (1986), The AQ15 inductive learning system: an overview and experiments, *in* 'Proceedings of IMAL 1986', Université de Paris-Sud, Orsay.

datasets	C4.5rules		CBA	OAC	
	NoDefault	Default	Default	NoDefault	Default
Anneal	59.4	88.2	66.5	89.1	91.3
Australian	56.1	74.9	69.6	72.7	77.6
Auto	51.1	62.4	64.0	71.1	72.3
Breast-cancer	61.6	87.7	89.9	92.2	93.1
Cleve	40.5	66.1	66.3	72.8	77.0
Crx	55.3	77.1	70.4	75.4	79.4
Diabetes	17.7	66.9	49.5	62.5	70.7
German	41.7	66.6	60.2	62.1	71.6
Glass	18.7	46.0	30.0	37.4	56.5
Heart	30.5	63.8	53.3	71.3	75.0
Hepatitis	61.4	81.3	77.9	75.8	81.8
Horse-colic	63.8	77.2	72.6	75.5	79.6
House-votes	68.3	82.8	82.9	87.9	90.3
Hypo	92.1	97.8	96.8	97.8	98.1
Ionoshere	77.3	91.0	93.1	91.8	93
Iris	0.0	21.3	38.7	0.0	24.7
Labor	33.2	80.5	79.4	67.3	80.5
Led7	3.9	13.5	12.1	15.4	22.1
Lymph	51.5	70.5	76.3	75.4	77.3
Mushroom	74.1	87.8	94.8	95.9	98.4
Pima	17.6	66.5	50.7	60.8	69.3
Sick	87.5	96.9	91.7	95.8	96.4
Sonar	53.5	74.8	75.6	75.2	77.7
Tic-tac	17.1	53.2	71.8	40.1	72.8
vehicle	36.1	51.4	61.9	65.6	67.5
Waveform	56.4	66.7	77.0	78.4	78.9
Wine	57.8	71.4	75.3	83.2	85.6
Zoo	61.0	72.1	72.5	64.6	76.5
Average	48.0	69.9	68.6	69.7	76.2

Table 6: Accuracy for test data with four missing values

Mingers, J. (1989), 'An empirical comparison of selection measures for decision tree induction', *Machine Learning* **3**, 319–342.

Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Rissanen, J. (1983), 'A universal prior for the integers and estimation by MDL', *Ann. of Statistics* **11**(2), 416–431.

Yin, X. & Han, J. (2003), CPAR: Classification based on predictive association rules, *in* 'Proceedings of 2003 SIAM International Conference on Data Mining (SDM'03)'.